References

bit.ly/JRChange
First, a story
Tell me about your title

Test Manager
- Focus is testing
- Organizing work around testing

Quality Leader
- Focus is whole life-cycle & process that leads to software delivery
- Influencing outcomes
Most Changes Efforts Fail

Why changes fail


1. Not establishing a great enough sense of urgency
2. Not creating a powerful enough guiding coalition
3. Lacking a vision
4. Under communicating the vision by a factor of 10
5. Not removing obstacles to the new vision
6. Not systematically planning, creating, short-term wins
7. Declaring victory too soon
8. Not anchoring changes into corporate cultures
Change is Hard

The Change Curve

Kübler-Ross model

1. Denial
2. Anger
3. Bargaining
4. Depression
5. Acceptance

Ref: MindTools, DREC, etc.
Tale of 2 changes

• Process / SDLC Change
  • Moving from 6-weeks to monthly release cycles

• Technology Change:
  • Institute static analysis in our development methodology
Roles in Change Leadership

• Change Leader
  • Leads in generating the vision
  • Drives the overall process for change

• Change Agent
  • Active Participant
  • “First Follower”
  • Video
Change leadership in 4 steps

1. Build the case for change
2. Plan the change
3. Test the change
4. Rollout and make adjustments
Build the case for change

• Identify the need
  • Retrospectives
  • Benchmark other organizations
  • Defect escapes / escalations
  • Industry trends / best practices
  • Ideas from team, innovations, experiments
  • Root cause analyses
  • Risk analysis
Build the case for change
Example: Defects found in system test

- Root cause analysis
  - 70% of errors found in system test were caused by coding errors
- How can we improve? (brainstorm)
  - Idea: start doing static analysis to find errors earlier
Build the case for change

• Think about your audience
  • Developers
  • Testers
  • Product Managers / Product Owners
  • Project Managers / Scrum Masters
  • Leaders
Communicate in the language of your audience

- Ideas -> Vision, Features -> Benefits

MP3 player
With small HDD

1000 songs
In my pocket

Faster release cycles
Quicker feedback
From customers

Static analysis checks
As part of the build

Automated code-
Review buddy
Plan the change

• Communicate, communicate, communicate
  • Organization’s goals
  • 1:1 sessions, Team meetings, all-hands
• Identify people who are:
  • Impacted by the change
  • Change agents
  • Passionate defenders of the status quo
• Ask:
  • What needs to change to make the vision reality?
  • What is working that we need to preserve?
  • What worries you?
The Goals Grid

- **Achieve**
- **Preserve**
- **Avoid**
- **Eliminate**

**Desirable**

- Things we have

**Undesirable**

- Don’t Have
- Things we have
## The Goals Grid

<table>
<thead>
<tr>
<th>Achieve</th>
<th>Preserve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firm completion milestone</td>
<td>Demos with support team before release</td>
</tr>
<tr>
<td>Regression in only 1 week</td>
<td>Staged deployment with subset of customers</td>
</tr>
<tr>
<td>Automated deployment</td>
<td>Feature configuration switches</td>
</tr>
<tr>
<td>Definition of Done =</td>
<td>Level of documentation</td>
</tr>
<tr>
<td>“Customer ready”</td>
<td>(requirements, design, etc.)</td>
</tr>
<tr>
<td>Better content</td>
<td></td>
</tr>
<tr>
<td>communication across</td>
<td></td>
</tr>
<tr>
<td>functions (ops, support, etc.)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Avoid</th>
<th>Eliminate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keeping the same practices, only compressing time and making people work OT</td>
<td>“This feature must release on this date” mindset</td>
</tr>
<tr>
<td>Embarrassing bugs</td>
<td>Late scrambles for design activities.</td>
</tr>
<tr>
<td>Double the manual</td>
<td>Pre-prod test environments instability</td>
</tr>
<tr>
<td>deployments across data</td>
<td></td>
</tr>
<tr>
<td>centers</td>
<td></td>
</tr>
<tr>
<td>Using “looming deadlines” to encourage urgency</td>
<td></td>
</tr>
</tbody>
</table>

**Don’t Have**

**Things we have**
Communicate, Communicate, Communicate, Communicate
Break out the scientific method

The Scientific Method as an Ongoing Process

- Make Observations: What do I see in nature? This can be from one's own experiences, thoughts, or reading.
- Think of Interesting Questions: Why does that pattern occur?
- Formulate Hypotheses: What are the general causes of the phenomenon I am wondering about?
- Develop Testable Predictions: If my hypothesis is correct, then I expect a, b, c, ...
- Refine, Alter, Expand, or Reject Hypotheses
- Gather Data to Test Predictions: Relevant data can come from the literature, new observations, or formal experiments. Thorough testing requires replication to verify results.
- Develop General Theories: General theories must be consistent with most or all available data and with other current theories.
## Test the change

<table>
<thead>
<tr>
<th></th>
<th>Faster releases</th>
<th>Static Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testable prediction</td>
<td>We can complete system test in 1 week if we use the new milestone definition</td>
<td>A tool exists that will find valid/meaningful coding errors with minimal false positives</td>
</tr>
<tr>
<td>Experiment</td>
<td>Use new milestone, keep 2 weeks in schedule for system test, but try to complete in 1</td>
<td>Identify a list of tools, and try out the most promising. Compare results</td>
</tr>
<tr>
<td>Refine</td>
<td>Locks on change control</td>
<td>Found 2 tools that worked well together (complementary results)</td>
</tr>
</tbody>
</table>
Rollout and Make Adjustments

Högertrafikomläggningen - Dagen H (Sept 3rd, 1967)
Be Persistent

- Actively seek out risks and issues
- Communicate openly about risks & progress (often)
- Ask for feedback: How can we improve this plan?
Show progress

Defects Detected by Static Analysis

Month 1  Month 2  Month 3  Month 4  Month 5  Month 6
Summary

• Building the burning platform is most important
  • Speak the language of your stakeholders
  • Articulate your vision
  • “be in love with the problem, not your solution”

• Communicate, Communicate, Communicate
  • Before, During, and After

• Persist when it gets difficult
  • Are you still solving the original problem?
    • Otherwise, pivot to a different solution
Thank You!

About Me
@JohnRuberto
blog.ruberto.com
linkedin.com/in/ruberto

Please leave a feedback card

Links to materials referenced are here: bit.ly/JRChange

About Concur
concur.com/careers
Travel & Expense Management w/ 35m active users
In top 50 best places to work (glassdoor.com)
Image Credits

• Slide 2: yonamaro from wikimedia
• Slide 9: US Coast Guard (public domain)
• Slide 13: (jobs) segagman from flicker creative commons
• Slide 13: (code) Pixabay (public domain)
• Slide 17: McDonalds Inc. (public domain)
• Slide 20: Jan Collsioo (public domain)
• Slide 24: Concur, Inc. (with permission)